

# **LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA AMBIENTE DE PROGRAMAÇÃO**

Prof. Dr. Daniel Caetano

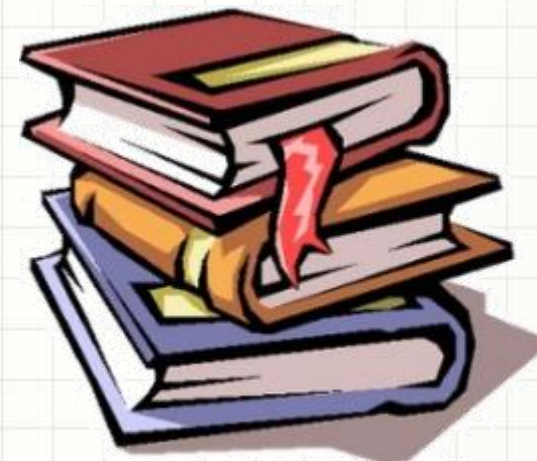
2013 - 2

# Objetivos

- Entender como avaliar divisibilidade
- Conhecer as funções matemáticas prontas do C/C++
- Capacitar o aluno para criar algoritmos sequenciais
- **Lista Aula 6!**



# Material de Estudo



---

## Material

## Acesso ao Material

Notas de Aula

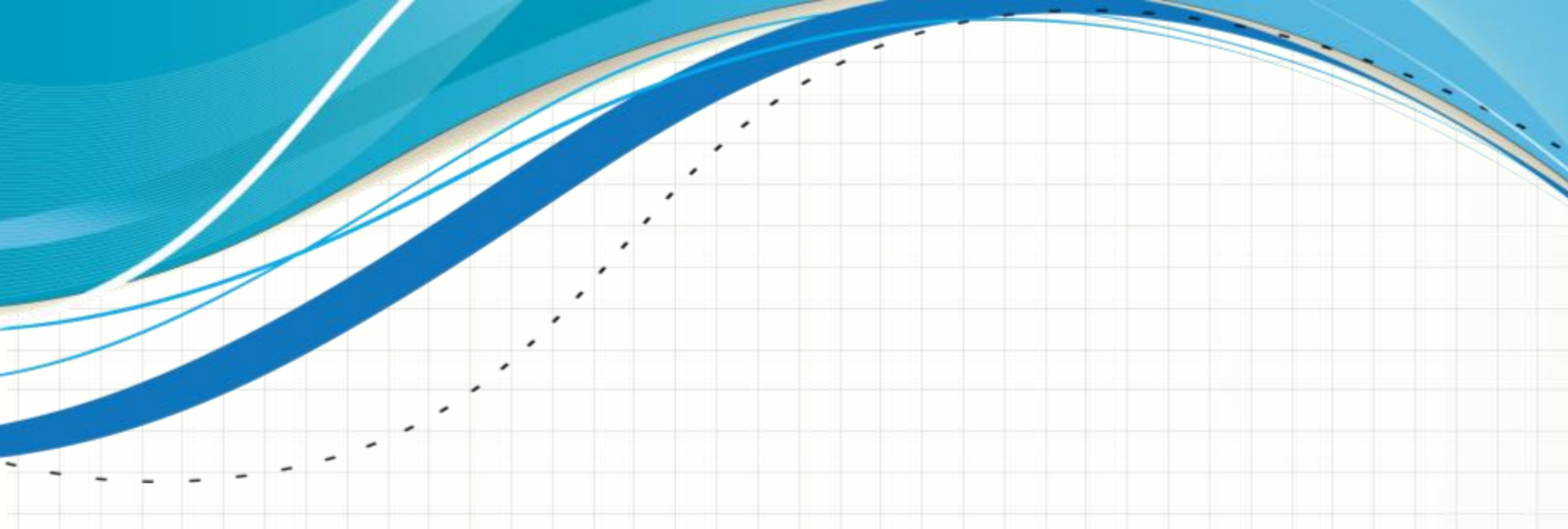
<http://www.caetano.eng.br/>  
(Lógica de Programação – Aula 6)

Apresentação

<http://www.caetano.eng.br/>  
(Lógica de Programação – Aula 6)

Material Didático

Lógica de Programação – Fundamentos da  
Programação de Computadores, páginas 7 a 47.



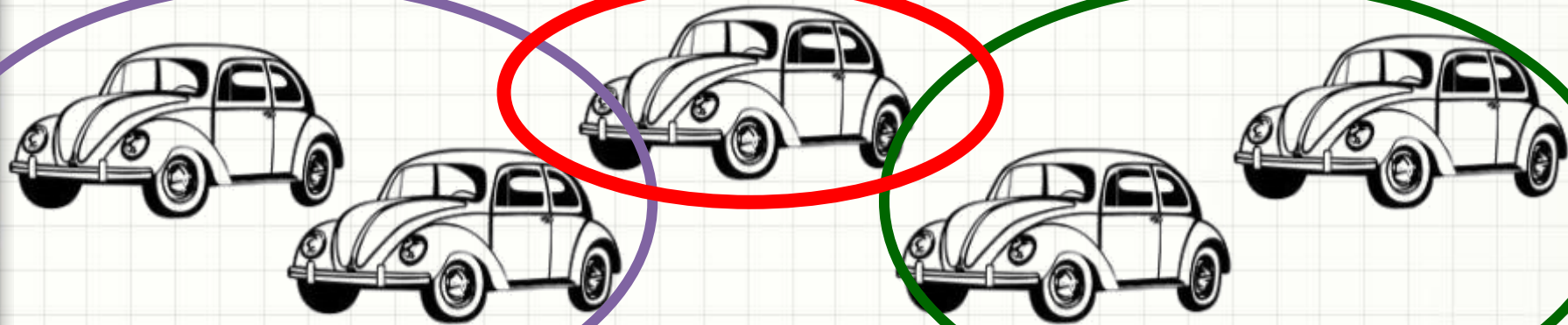
# **NÚMEROS DIVISÍVEIS**

# Número Par ou Ímpar?

- Como determinar se um número é par?
- Par: divisível por dois
- O que significa ser divisível por 2?



# Divisível por Dois



# Resto de Divisão

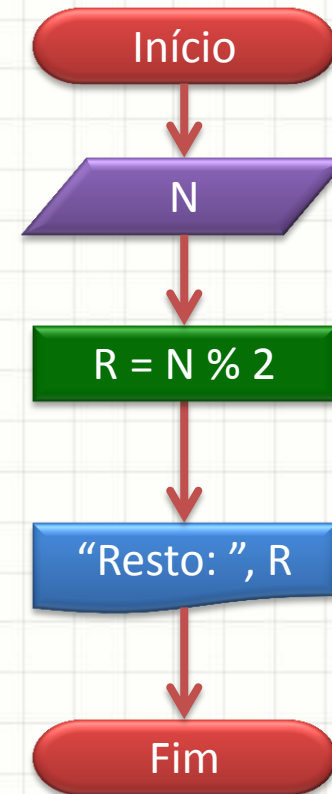
- Como determinar se um número é par?
- Par: divisível por dois
- O que significa ser divisível por 2?
- Significa que o resto da divisão por 2 é 0!
- Vamos experimentar:
  - Algoritmo que imprime resto da divisão por 2

# Verificando Paridade de um N<sup>o</sup>

- Linguagem Natural

1. Leia um número
2. Calcule o resto da divisão por 2
3. Imprima o resto

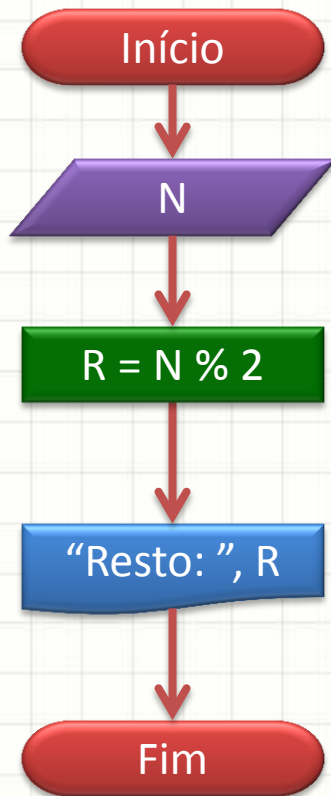
- Fluxograma





# Verificando Paridade de um N<sup>o</sup>

- Fluxograma



- Portugol

**Algoritmo** "Calcula Paridade"

**Inicio**

**Inteiro** N, R

**Escreva** "Digite um No.:"

**Leia** N

$R \leftarrow N \% 2$

**Escreva** "Resto:", R

**FimAlgoritmo**

# Verificando Paridade de um N<sup>o</sup>

- Portugol

**Algoritmo** “Calcula Paridade”

**Inicio**

Inteiro N1, R

Escreva “Digite um No.:”

Leia N1

$R \leftarrow N \% 2$

Escreva “Resto:”, R

**FimAlgoritmo**

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main() // Calcula Paridade
```

```
{
```

```
    int N, R;
```

```
    cout << “Digite um No.:”;
```

```
    cin >> N1;
```

```
    R = N % 2;
```

```
    cout << “Resto: ” << R;
```

```
}
```

# Verificando Paridade de um N<sup>o</sup>

- Portugol

- Linguagem C

**#include <iostream>**

**Como imprimir “Par” se o número é par e “Ímpar” se o número é ímpar?**

**Algoritmo**

**Início**

**Inteiro**

**Escreva**

**Leia N1**

**$R \leftarrow N \% 2$**

**Escreva “Resto:”, R**

**FimAlgoritmo**

**cin >> N1;**

**R = N % 2;**

**cout << “Resto: ” << R;**

**}**



# **CONVERSÃO PARA UNIDADES NÃO DECIMAIS**

# Conversão de Segundos para M:S

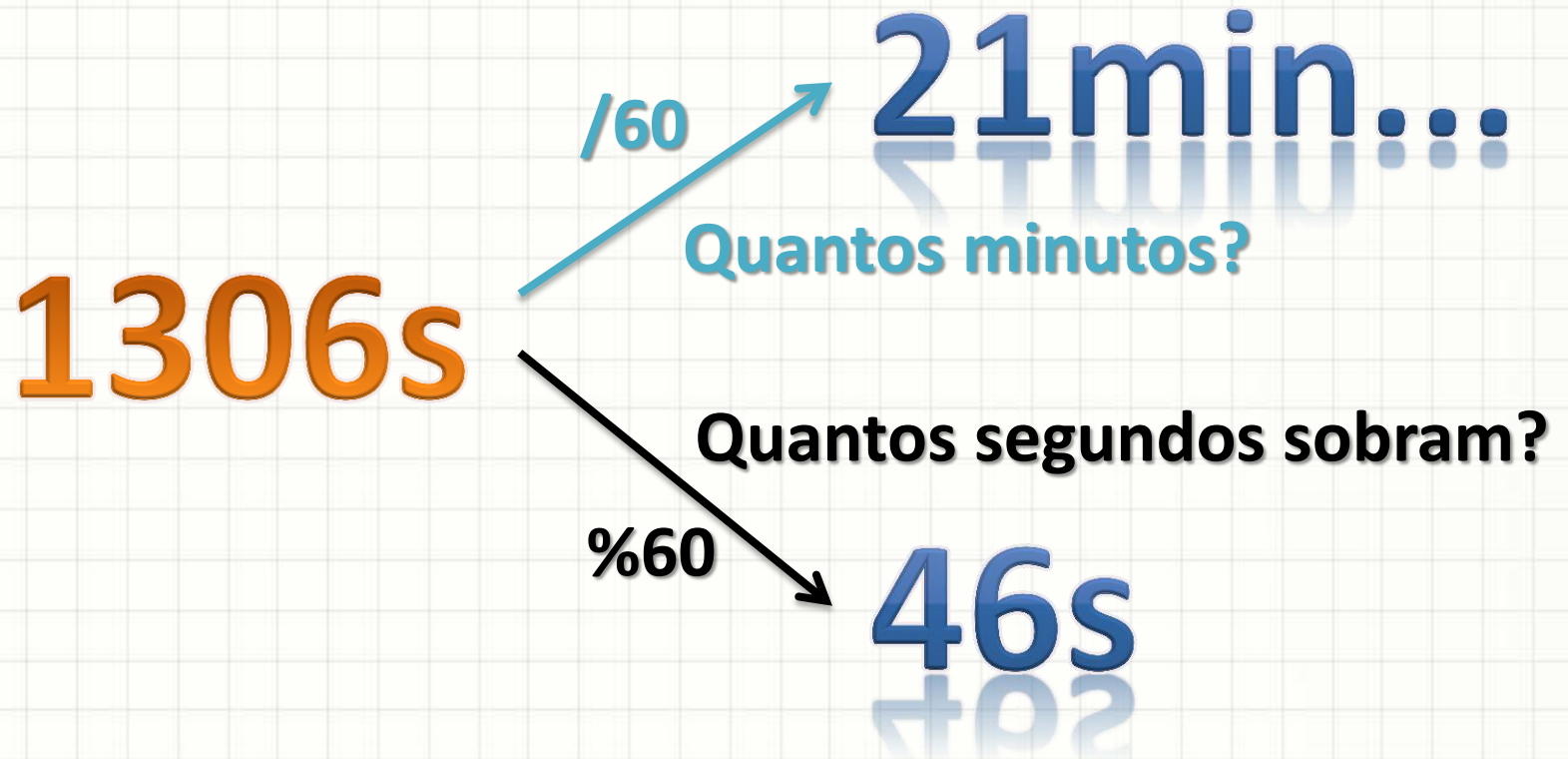
- Convertendo 1346 segundos em
  - Minutos e segundos

1346

M : S



# Conversão de Segundos para M:S



Vamos Sistematizar!

# Conversão de Segundos para M:S

- Convertendo 1.306s  $\rightarrow$  M : S
- 1min  $\rightarrow$  60s
- Quantos minutos tem em 1.306s?

$$1.306 / 60 = 21,7666666... \text{ minutos}$$

**21 : S**

- 21 minutos e “uns quebrados”...
- Quantos segundos sobraram?

# Conversão de Segundos para M:S

- Convertendo 1.306s  $\rightarrow$  21 : S
- 1min  $\rightarrow$  60s
- Qtos segs. não completam 1min. em 1.306s?

$$1.306 \% 60 = 46... \text{ segundos}$$

**21 : 46**

# Conversão de Segundos para M:S

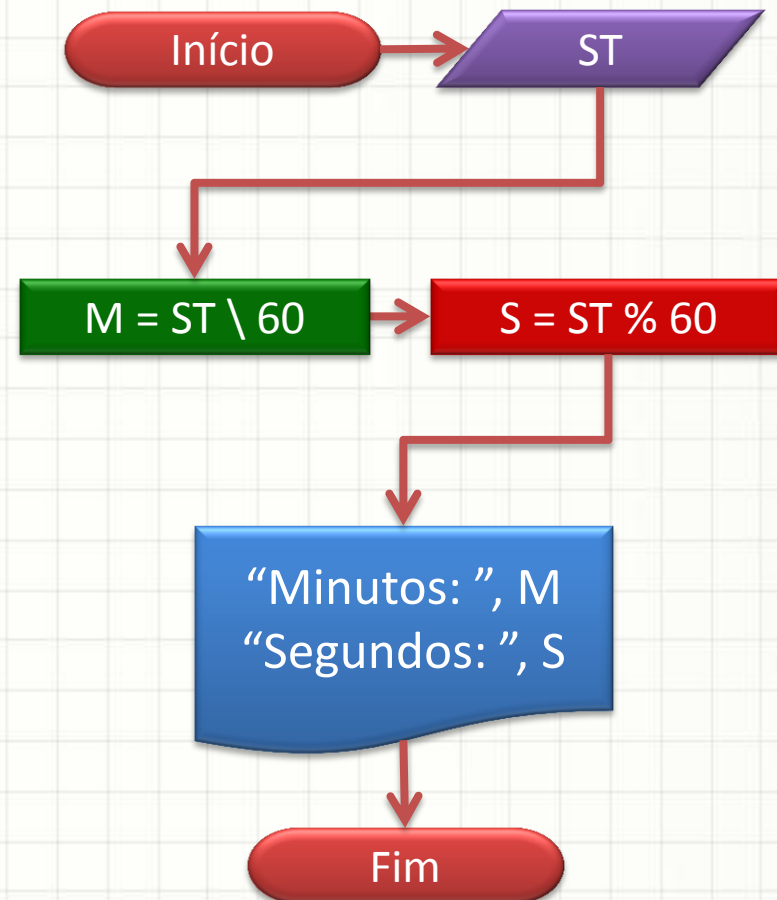
- Resumindo
  - $1.306 / 60 = \mathbf{21,76666666...}$  minutos
  - $1.306 \% 60 = \mathbf{46}$  segundos (sobram)
- Assim:
  - $1.306s = 21min, 46s$
- Vamos representar isso como um algoritmo?

# Conversão de Segundos para M:S

- Linguagem Natural

1. Leia o número de segundos totais
2. Calcule os “minutos”, dividindo os “segundos” por 60 (divisão inteira)
3. Calcule os “segundos” restantes, com o resto de divisão por 60
4. Imprima o número de minutos e segundos

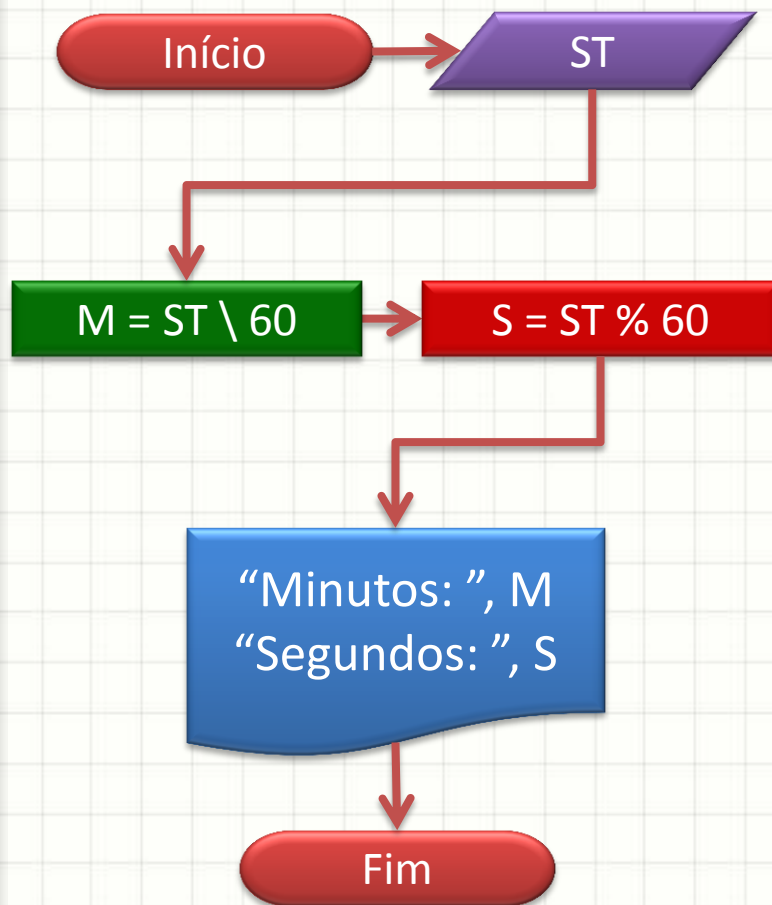
- Fluxograma





# Conversão de Segundos para M:S

- Fluxograma



- Portugol

**Algoritmo** "Segundos para M:S"

**Início**

**Inteiro** ST, M, S

**Escreva** "Quantos segundos? "

**Leia** ST

$M \leftarrow ST \div 60$

$S \leftarrow ST \% 60$

**Escreva** "Minutos: ", M, **fiml**

**Escreva** "Segundos: ", S, **fiml**

**FimAlgoritmo**

# Conversão de Segundos para M:S

- Portugol

**Algoritmo** “Segundos para M:S”

**Inicio**

**Inteiro** ST, M, S

**Escreva** “Quantos segundos? ”

**Leia** ST

$M \leftarrow ST \div 60$

$S \leftarrow ST \% 60$

**Escreva** “Minutos: ”, M, **fiml**

**Escreva** “Segundos: ”, S, **fiml**

**FimAlgoritmo**

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main() // Segundos para M:S
```

```
{
```

```
    int ST, M, S;
```

```
    cout << “Quantos segundos? ”;
```

```
    cin >> ST;
```

```
    M = ST / 60;
```

```
    S = ST % 60;
```

```
    cout << “Minutos: ” << M << endl;
```

```
    cout << “Segundos: ” << S << endl;
```

```
}
```

# Conversão de Segundos para M:S

- Portugol

**Algoritmo** “Seg

**Inicio**

**Inteiro** ST, M

**Escreva** “Qu

**Leia** ST

$M \leftarrow ST \div 60$

$S \leftarrow ST \% 60$

**Escreva** “Minutos: ”, M, fiml

**Escreva** “Segundos: ”, S, fiml

**FimAlgoritmo**

- Linguagem C

**#include** <iostream>

**using namespace** std;

**Na apostila há uma  
versão completa, para  
H:M:S**

**Tente entender o processo!**

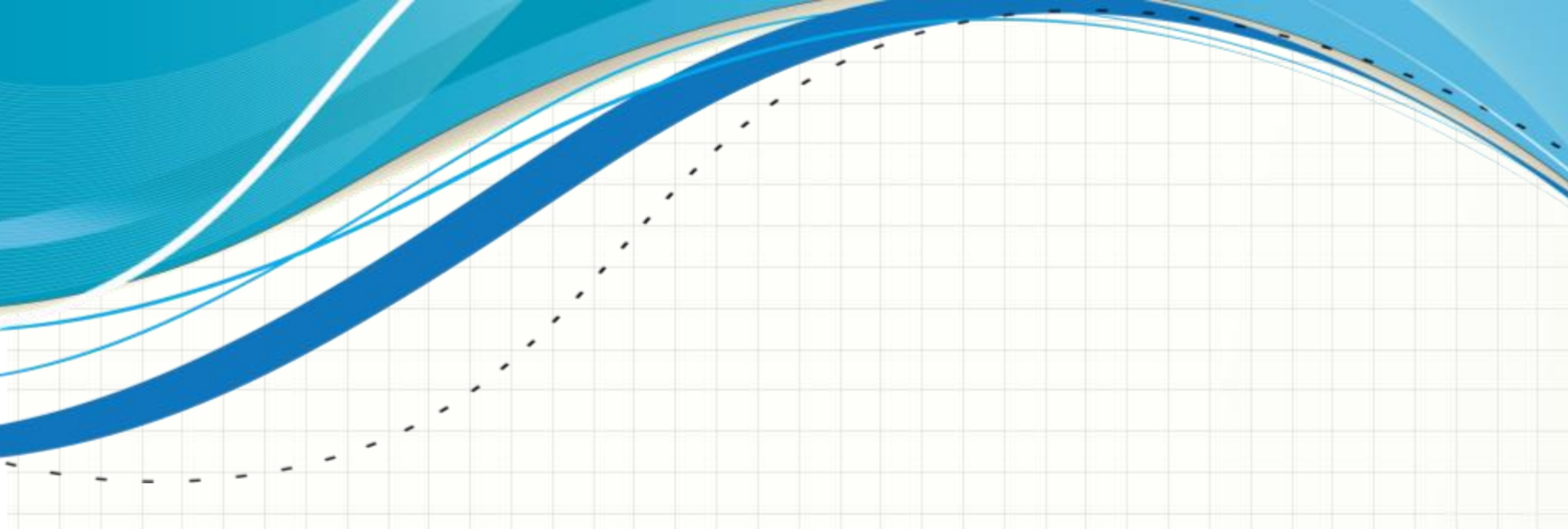
**egundos? ”;**

$S = ST \% 60;$

**cout** << “Minutos: ” << M << **endl;**

**cout** << “Segundos: ” << S << **endl;**

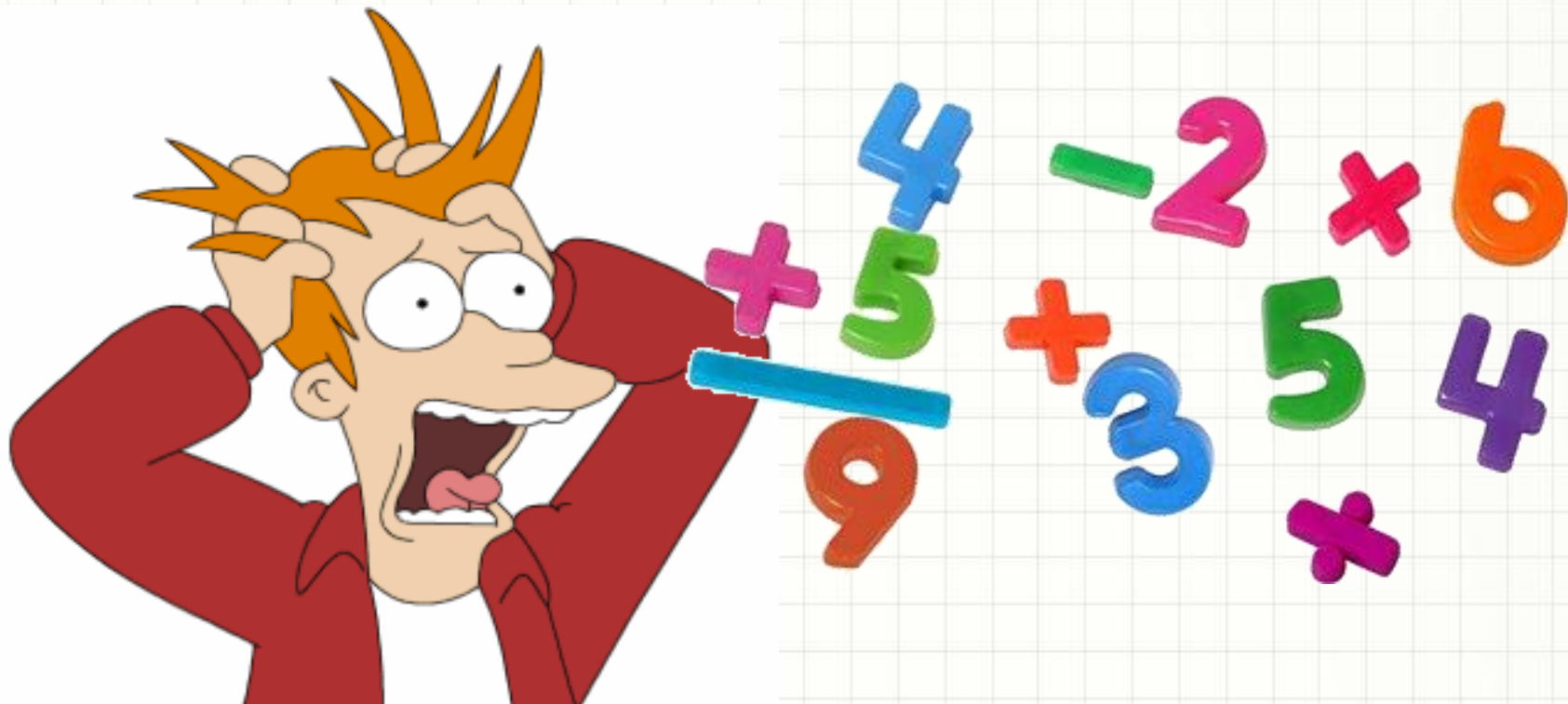
**}**



# **FUNÇÕES MATEMÁTICAS**

# Funções Matemáticas

- Vimos somas, subtrações, multiplicações...
  - Mas o computador não faz cálculos complexos?
  - Cadê o logaritmo, a raiz quadrada etc.?



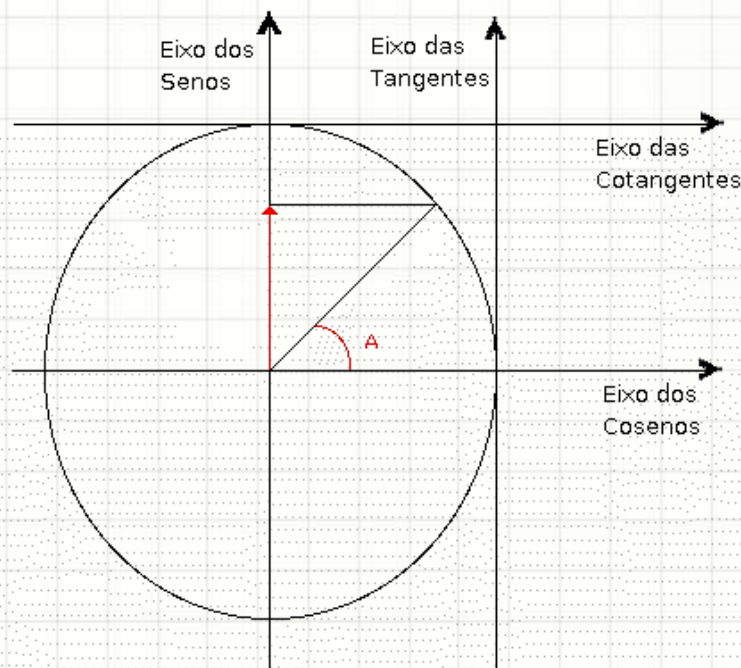


# Funções Matemáticas

- Esses “caras” são chamados de **funções**
- **O que é uma função?**

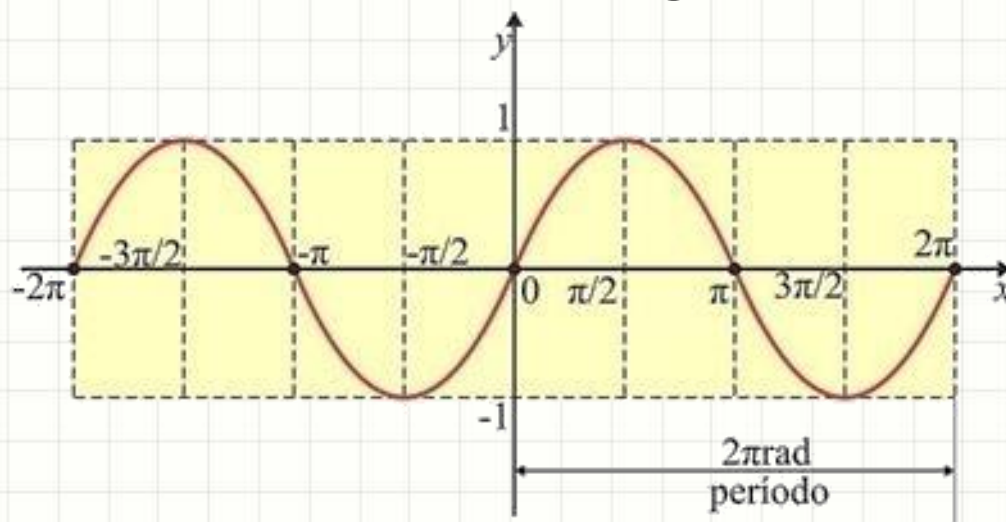
## Qual é o seno?

- Seno de quê?



# Funções Matemáticas

- O “seno” depende do valor do ângulo
  - Seno é calculado “**em função**” do ângulo
  - Ex.: Qual o seno do ângulo  $60^\circ$ ?



- O mesmo vale para cosseno, tangente, logaritmo e tantas outras!

# Funções Matemáticas

- Em C/C++, uma função é como se fosse...
  - Um novo comando
  - Uma nova tarefa que o computador sabe executar
- Existem várias funções prontas no C/C++
- **Elas são organizadas em bibliotecas**
  - No caso, falamos da biblioteca **matemática**

**#include <math.h>**

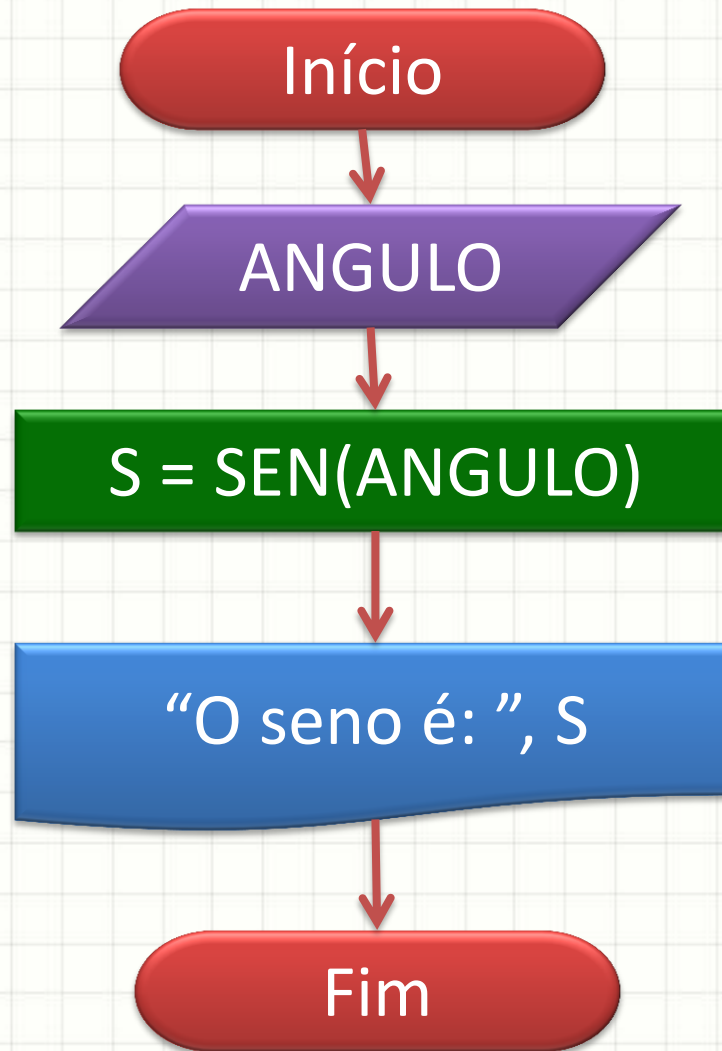
# Funções Matemáticas

C/C++	Função
abs(x)	Devolve o valor absoluto (sem sinal) de <b>x</b>
round(x)	Devolve <b>x</b> arredondado
log(x)	Devolve o logaritmo natural de <b>x</b>
pow(x,y)	Devolve o valor de <b><math>x^y</math></b>
exp(x)	Devolve o valor de <b><math>e^x</math></b>
sqrt(x)	Devolve o valor da raiz quadrada de <b>x</b> (SQuare RooT)
sin(x)	Devolve o valor do seno de <b>x</b> (radianos)
cos(x)	Devolve o valor do cosseno de <b>x</b> (radianos)
tan(x)	Devolve o valor da tangente de <b>x</b> (radianos)
M_PI	Representa o valor de PI (3,141592...)

# Exemplo: Calculando o Seno

- Como um exemplo, vamos calcular o seno de um ângulo
1. Vamos ler um ângulo
  2. Vamos calcular o seno
  3. Vamos imprimir o seno

# Exemplo: Calculando o Seno





# Exemplo: Calculando o Seno

**Algoritmo** “Calcula o seno de um ângulo”

**Inicio**

**Real** ANGULO, S

**Escreva** “Digite um ângulo – 0 a  $2\pi$ : ”;

**Leia** ANGULO

**S**  $\leftarrow$  **sen**(ANGULO)

**Escreva** “Seno: ”, S, **fiml**

**FimAlgoritmo**

# Exemplo: Calculando o Seno

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
main() // Calcula o seno de um ângulo
```

```
{
```

```
    float ANGULO, S;
```

```
    cout << "Digite um ângulo – 0 a 2*PI: ";
```

```
    cin >> ANGULO;
```

```
    S = sin(ANGULO);
```

```
    cout << "Seno: " << S << endl;
```

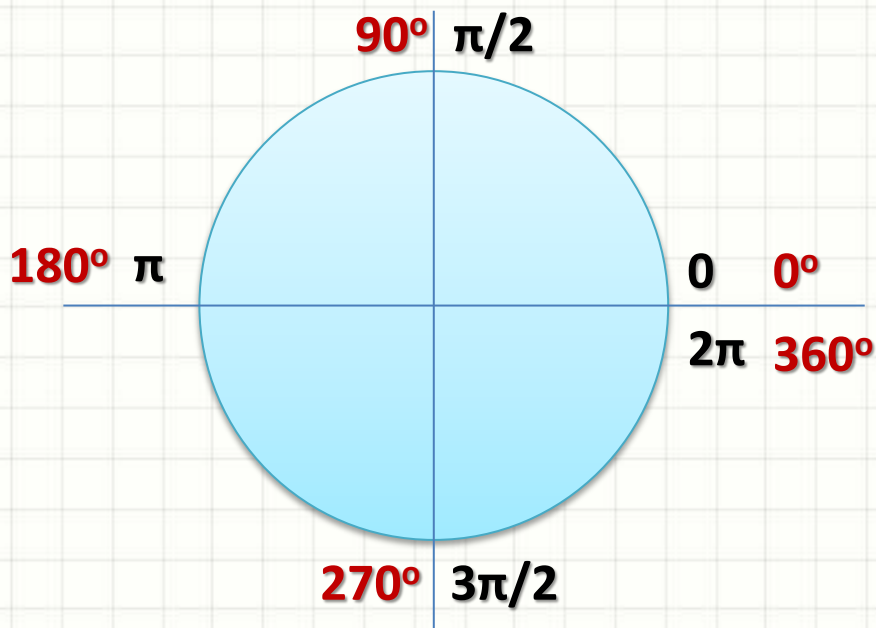
```
}
```

# Exemplo: Calculando o Seno

- E se quisermos ler o ângulo em GRAUS?
  1. Leia um ângulo (em graus)
  - 2. Converta-o em radianos**
  3. Calcule o seno do ângulo em radianos
  4. Imprima o valor do seno
- Como converter AR em AG?

# Exemplo: Calculando o Seno

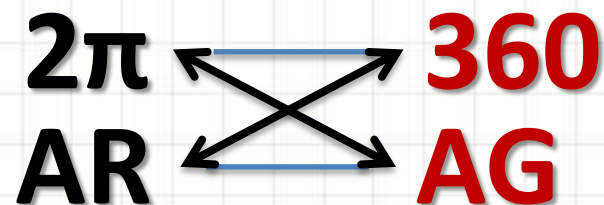
- Convertendo ângulo em graus para radianos



$$\begin{array}{ccc} 2\pi & \text{—} & 360 \\ \text{AR} & \text{—} & \text{AG} \end{array}$$

# Exemplo: Calculando o Seno

- Convertendo ângulo em graus para radianos

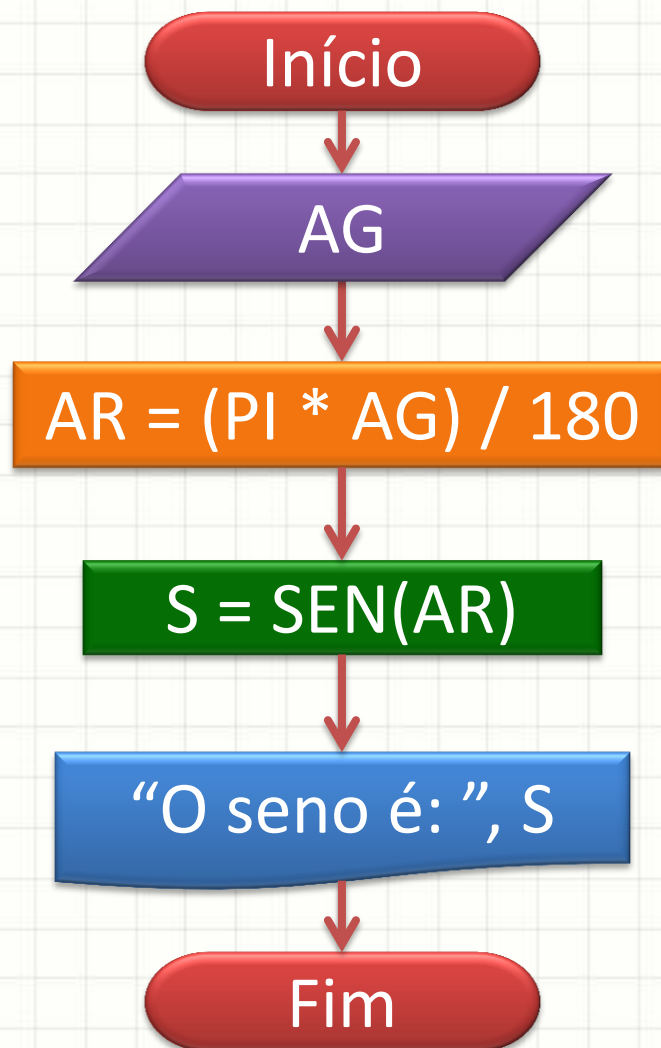

$$\begin{array}{cc} 2\pi & \longleftrightarrow & 360 \\ AR & \longleftrightarrow & AG \end{array}$$

$$AR * 360 = 2 * \pi * AG$$

$$AR = (2 * \pi * AG) / 360$$

$$AR = (\pi * AG) / 180$$

# Exemplo: Calculando o Seno





# Exemplo: Calculando o Seno

**Algoritmo** “Calcula seno de ângulo em graus”

**Início**

**Real** AG, AR, S

**Escreva** “Digite um ângulo – 0 a 360: ”

**Leia** AG

$AR \leftarrow (AG * PI) / 180$

$S \leftarrow \text{sen}(AR)$

**Escreva** “O seno é: ”, S, **fiml**

**FimAlgoritmo**

# Exemplo: Calculando o Seno

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
main() // Calcula o seno de um ângulo em graus
```

```
{
```

```
    float AG, AR, S;
```

```
    cout << "Digite um ângulo – 0 a 360: ";
```

```
    cin >> AG;
```

```
    AR = (M_PI * AG) / 180.0;
```

```
    S = sin(AR);
```

```
    cout << "O seno é: " << S << endl;
```

```
}
```



# ARREDONDAMENTO

# Função de Arredondamento

- Como arredondar um valor?

**float** valor, arredondado;

valor = 1.55;

arredondado = **round**(valor);

- Qual o valor de **arredondado**?
- Arredondamento em C: função limitada!
  - Vamos ver um exemplo!

# Arredondando Números - C

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
main() // Arredonda um número
```

```
{
```

```
    float NUM, ARRED;
```

```
    cout << "Digite um número fracionário: ";
```

```
    cin >> NUM;
```

```
    ARRED = round(NUM);
```

```
    cout << "Arredondado para: " << ARRED << endl;
```

```
}
```

# Função de Arredondamento

- **Para pensar:**

- Como fazer, usando apenas aritmética básica e a função **round**, para:
  - Arredondar com 1 casa decimal?
  - Arredondar com 2 casas decimais?
  - Arredondar com 3 casas decimais?
  - ...





# CONCLUSÕES

# Resumo

- Resto de Divisão
    - Fracionar números em unidades menores
    - Verificar divisibilidade
  - C/C++: vários cálculos matemáticos prontos
  - **TAREFA: Lista Aula 6!**
- 
- Vamos exercitar um pouco?
    - Praticar compreender e sistematizar!



**PERGUNTAS?**